A Mobile Communication Device for a Safety-Critical Domain: From Object-Orientation to HCI Design

Christian M. Nielsen, Michael Overgaard, Michael B. Pedersen, Sigge Stenild, and Jan Stage

Aalborg University, Department of Computer Science,
Fredrik Bajers Vej 7, DK-9220 Aalborg East, Denmark
{monrad, miov, mbp, stardust, jans}@cs.auc.dk

Abstract: Safety-critical domains are interesting and challenging areas for application of mobile devices. This paper presents results from an experimental evaluation of an object-oriented method for analysis and design of mobile systems. This method provides guidelines for the design of interactive mobile systems based on object-oriented models of the application area. The evaluation involves design of a specific location-aware mobile device to support communication in a safety-critical domain. The domain is the fuel department of a coal-based power plant. It is concluded that the method provides valuable support to the design of the user interface. Yet the initial activities and the integration of prototypes in the development are not supported to the same extent.

Keywords: communication, safety-critical domain, object-oriented, interface design

1 Introduction

Mobile handheld devices, such as mobile phones and personal digital assistants (PDAs), are spreading fast and are becoming ever more powerful. The mobile technology has provided organizations, as well as individuals, with the ability to work in novel and previously unanticipated ways [25]. A variety of mobile devices are already being used in almost every workspace, to obtain information and interact with other users, especially for communicative purposes [13]. In addition, mobile devices are also used to participate in the social practices of exchange [4, 27].

When we develop systems for mobile devices, a variety of aspects that are different from those considered in relation to traditional systems, have to be taken into account [18]. Compared to traditional stationary desktop systems, mobile systems are unique in several ways: They are often used while moving, they are used in different locations and situations, and they often have very small visual displays, which result in limited interactions styles [8, 9, 16].

These fundamental differences impose new and fundamentally different challenges on the analysis and design of mobile systems. Especially the design of usable user interfaces leads the user interface designer to explore other ways of displaying information to the mobile user. In order to take all of these aspects into account in the development process, it requires a study of human-computer interaction (HCI) processes to support usability of the mobile devices [3]. There is a substantial amount of articles documenting HCI research in relation to mobile systems, but the area of mobile HCI lacks research and development based on a firm methodological foundation, since most of these activities are based on trial and error [17].

Methods are generally used with the purpose of solving a problem. A software development method usually consists of a series of activities, being abstract prescriptions on how to reach a state where the problem has been solved. The purpose of a method is to enable practitioners with different backgrounds to avoid repeating mistakes that others have encountered. Thereby, methods enable less experienced people to be more likely in succeeding [6]. Methods cannot be expected to solve all problems, but they can make the life of practitioners somewhat easier.

This paper presents results from an experiment where we studied to what extent an existing software development method supported the design of the HCI aspects of a mobile system. The following section 2 describes the experiment. This includes an overview of the methods that were considered as candidates for the experiment and the details of the experimental procedure that was applied. Section 3 presents the user organization where we conducted the experiment. The organization is a coal-based power plant, and the application domain is the work processes in the fuel department. Section 4 provides a description of the method that was used combined with experiences from the application of the method for analysis and design of a mobile handheld device to support communication in the safety-critical domain. In section 5, we discuss general lessons learned through the experiment. Finally, section 6 provides the conclusion.

2 Experimental Design

The purpose of the experiment was to study to what extent an existing software development method supported the design of the HCI aspects of a mobile system. The basic research method employed in the experiment was a case study approach. Case studies are often thorough empirical studies of small sized entities such as groups, organizations, individuals, or systems. Case studies are particularly well suited for research focusing on describing and explaining a specific phenomenon and for developing hypothesis or theory Kjeldskov & Graham [2003]. Our aim was gain more insight into the potentials of general object-object oriented methods for development of mobile systems.

2.1 Development Method

A key decision was the selection of the specific method that was used in the case study. In this selection, we limited ourselves to object-oriented methods. There are three reasons for this. First, object-oriented methods have become the most widely used approach in the development of interactive software systems [20]. Second, the basic concepts of the object-oriented approach, objects, states, and behaviours, are well suited for describing the system's context [20]. This relation to context is particularly important when we are developing mobile, location-aware systems. In addition, object models are central in analysing and understanding the problems relevant to the users. This includes the users' understanding of the system and their envisioned behaviour while using the system [11]. Third, the members of the development team had been using object-oriented methods and programming languages in other projects. Thereby, they had considerable experience with the object-oriented approach.

Within the object-oriented approach, we had to select a specific method. We identified a collection of articles about different methods that combine object-oriented analysis and design with HCI [11]. The methods considered are listed in the top row of table 1. In order to select one of these methods, we developed a list of aspects that a method might include. This list is the left column of table 1. The result of evaluating the methods in the collection is shown in table 1. A dot in the table indicates an area of focus in the article that presents the method.

Table 1. Comparison of methods

	Part I - Participatory design	**Part II** - Scenario - and task-based design			**Part III** - Use-case based design			**Part IV** - User centered design	
	Chapter 1 [McGinnes & Amos, 2001]	**Chapter 2** [Rosson & Carroll, 2001]	**Chapter 3** [Van Harmelen, 2001-2]	**Chapter 4** [Artim, 2001]	**Chapter 5** [Kruchten et al., 2001]	**Chapter 6** [Nunes & Cunha, 2001]	**Chapter 7** [Constantine & Lockwood, 2001]	**Chapter 8** [Gulliksen et al., 2001]	**Chapter 9** [Hudson, 2001]
Project									
Rich pictures	•			•					
Project Size	L	N/A	N/A	M-L	L	S	All	L	N/A
Application domain									
User profiles/Roles			•	•	•	•	•		•
Use-cases				•	•	•	•	•	•
Scenarios		•	•	•			•		•
Stakeholders									•
Problem domain									
Object modelling	•		•	•	•	•			
Examples of object modelling	•		•	•	•	•			
User interface									
Interface modelling	•			•	•	•			
Examples of interface modelling	•					•			
Other									
Based on	DSDM, RAD	OOA	Idiom94	ETP	RUP	UCEP, LUCID, OMT, Spiral Model, Bridge, OVID,		RUP, DSDM, ISO 13407	ISO 13407

Table 1 illustrates that some methods focus only on a few areas, while others cover several areas of the criteria. Based on this analysis, we chose Wisdom (Whitewater Interactive System Development with Object Models) [24] as the method we would use in our experiment. It covers criteria within all of the five main categories. Furthermore it includes the Bridge method as described by [7] in the final stages of the development process for describing the transition from object models to user interface design, and is thereby the only method that elaborates explicitly on this issue.

We were aware that neither Wisdom nor any of the other methods in table 1 cover all parts of the development process. Therefore, we decided to combine it with a general object-oriented method known as OOA&D [20]. The idea was to use this method as a supplement when we encountered a development process problem that was not covered by Wisdom.

2.2 User Organization

The other key decision was the choice of user organization. We are generally focussing on safety-critical domains. Therefore, we would prefer a user organization within such a domain. In addition, we stated a list of requirements to the case´. Based on these, We chose Nordjyllandsværket, see figure 1, which is coal-based power plant situated in Northern Jutland in Denmark. We limited the case to the fuel department of the power plant. This department is a safety-critical domain where a mobile system will be useful in different locations, and where the employees have different and varying working tasks to solve. Tasks are carried out in collaboration among employees, and during this process the employees must be able to communicate with each other, even if they are not located in the same place.

Figure 1. Nordjyllandsværket

2.3 Experimental Procedure

The experiment basically involved analysis and design of a mobile communication device for employees in the fuel department. The experiment would involve development of a series of prototypes that represented a specific design of the mobile system. After analysis and design, we would test a final prototype through a usability evaluation, since this is the most systematic way to discover, whether the method has succeeded in assisting our development of a usable system.

Data collection throughout the development process was based on diaries [15]. The diaries were used to retain important information, such as observations, actions, and reflections. The diaries later served as the basis for reflections on the development process. The development process was carried out by a development team that consisted of four of the authors of this paper. It was conducted during the fall of 2003.

3 A Safety-Critical Domain

This section provides an overview of the user organization in our development case. It is the fuel department (see figure 2) at the coal-based power plant Nordjyllands-værket, located in Northern Jutland, Denmark. This includes a description of the major tasks performed by the employees in the fuel department and the nature of the problems they experience with their collaboration.

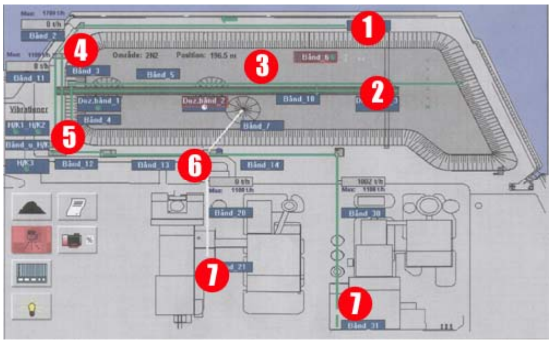

Figure 2. Overview of the fuel department

Nordjyllandsværket is a coal-based power and heating plant located in North Jutland in Denmark. The production of central heating and electricity result in several by-products such as sulphuric acid and ashes that are used in the production of cement. The plant is divided into two independent production plants. The coal to the two plants is supplied from a central storage area. The fuel-department is responsible for delivering the coal used in the two production plants, amounting daily to 5000 tons of coal for each.

3.1 Communication to Support Coordination

The employees perform a variety of different tasks in order to ensure that the needed amount of coal is delivered to the two production plants. In order to coordinate the many tasks described above the need for quick and easy communication is important, and in some cases even essential in order to carry out the job in a safe and efficient manner. At the present, the devices used for communicating are VHF-radios (walkie-talkies), DECT wireless phones, and some times mobile phones.

Every element of the coal transport can be controlled through the existing system. When a problem arises, which cannot be solved from the control room, for example in the Grinder building, the person situated here trying to solve a problem do not have access to the information, which is available in the control room. Furthermore only some parts of the machinery can be controlled from the Grinder building. The only way to gain some kind of access to the information systems in the control room is by communicating with a person in the control room either by phone or walkie-talkie. Often the phones are not usable because of the weak signal, hence the only tool for communicating is the walkie-talkie, but this is problematic if the machine is running, since the Grinder generates a lot of noise.

3.2 Communication Problems

Several problems are related to the use of these devices: Many conveyor-belts run underground, which disrupts the signal, and the machines and conveyor belts are often placed inside concrete buildings, which also disrupts the signal. Finally, there is a deafening noise inside these buildings, which makes talking to each other difficult, and using some kind of mobile devices for verbal communication is virtually impossible.

In their experience the problems were related to two areas. One area was explained as problems with connection fallouts when using their communication devices. The second area was problems with noise in some areas of the power plant. We also had a tour around the work area of the fuel department to gain geographical knowledge of the application domain.

4 Development Method and Process

In this section we describe the Wisdom method's development process and our actual development process. Wisdom is is described in different references [22, 23, 24].

The Wisdom process is a software process framework based on a user-centered, evolutionary, and rapid-prototyping model specifically adapted for small teams of developers. The process includes four phases of software development, which are called "Evolutionary phases", because of Wisdoms evolutionary approach: Inception, Elaboration, Construction, and Transition. These phases are included with different intensity in the four workflows of the method: Requirements, Analysis, Design, and Whitewater evolution. The fourth workflow is named Whitewater evolution, and replaces traditional approaches to implementation. Evolution complements whitewater because the process evolves in a sequence of incremental prototypes.

Our objective was to evaluate the Wisdom method. Therefore, we wanted to use it to the extent possible, making sure that all activities had been performed before we turned to the alternative method. Figure 3 illustrates what we actually used Wisdom for and how we combined it with our alternative method.

The first workflow deals with requirements. It quickly turned out that it was not clear how to initiate the development process. We conducted studies to obtain an overview of the application domain. This gave us an overview of the application domain and insight to their communication problems, using wireless phones and walkie-talkies in such a noisy environments. The understand system context activity is conducted for understanding the domain of the problem ending up with the domain model. The domain model captures the most central type of objects in the context of the system. The user profiling is conducted in order to describe the actual users whose tasks will be supported by the system. The last activity of the requirement workflow is essential task modelling. It aims at finding the functional and non-functional requirements of the envisioned system. Wisdom is an essential use-case and task flow based method, and therefore it relies on essential use cases to capture the structure of use patterns and the underlying functional requirements.

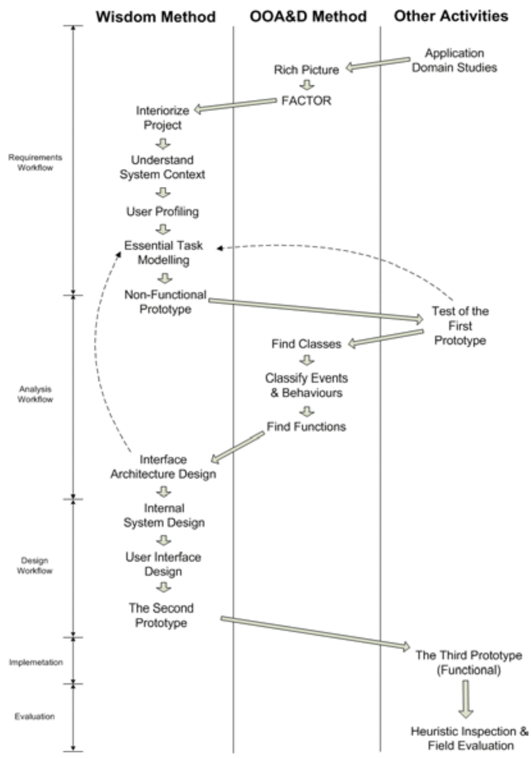

Figure 3. The development process

To evaluate the outcome of the requirement workflow, a non-functional prototype was created (prototype 1). The Wisdom method did not describe which approach to use, when testing the non-functional prototype. Therefore we did an informal test at the fuel department, where different employees tried to use the prototype, which led to a discussion about the required functionality and the prototype's structural design. The outcome of the test, resulted in another re-evaluation of the task flow diagrams, and the test yielded useful ideas that should be considered when designing the next prototype.

The analysis workflow refines the structure of the system that was described in the requirement workflow. The purpose is to produce a description of the requirements that shapes the structure of the interactive system. One outcome of this activity is a revised class diagram, but since the Wisdom method did not provide any detailed information on the procedure, we decided to use OOA&D to find classes, events, behaviours, and functions, and to produce the required revised class diagram. The analysis workflow also includes interface architecture design and construction of an interaction model. The interaction model is the external view of the system from the user interface perspective. This model structure the user interface, and therefore identifying the different elements that compose the dialogue and presentation structure of the system, and how they relate to the domain-specific information in the functional core.

Figure 4. The user interface of the communication device

In the design workflow the shape and architecture of the system are refined for the implementation. The internal system design contains two sub-activities, where the first is to prioritise and select use cases, and the second is to design the use case classes. In the user-interface design activity, the interaction model is transformed into a concrete user interface design. This activity encompasses two concurrent flows of activities, corresponding to the dialogue and presentation components of the interaction model. The user-interface design activity ends by relating both task classes to interaction space classes, hence completing the process of distributing responsibilities between the dialogue and presentation models.

The Wisdom method refers to the Bridge method for detailed interface design. The Bridge method is a comprehensive and integrated methodology for designing an object-oriented, multi-platform, graphical user interfaces with the objective of meeting user needs [7]. The Bridge method was used to map the interaction spaces into a concrete user interface design, which resulted in another non-functional prototype (the second prototype). The non-functional prototype was used as an outline for the implementation of a functional prototype (the third prototype), which was an iterative activity that concludes the Wisdom design workflow. The resulting interface is illustrated in figure 4.

5 Evaluation of the development method

In this section we will focus on the activities in the requirement, analysis, and design workflow of the Wisdom method and discuss and evaluate problems and successes experienced. This evaluation is based on the diaries made during the development process.

5.1 Workflows

In the requirement workflow there was a general problem of just getting started. To solve this problem, two OOA&D techniques were included. Another problem was capturing the communication between the employees. To solve some of the challenges, a situation was staged, in which we were able to observe the communication, and through a non-functional prototype we were able to check whether our understanding of the communication structure was acceptable. Through Wisdom's user profiling activity, we were supposed to find the actors of the system, but in doing this, we ended up with the question of what is the difference between the terms actors and roles and what constitutes a user profile?

In the analysis workflow, we chose to include techniques from the OOA&D method for finding classes, events, behaviours and functions in the system. The outcome of using these techniques was a revised class diagram that depicted the physical locations, the communication, and the users relevant to our system.

In the design workflow we started out by re-evaluating the use cases and essential task flow diagrams, and based on this, the interaction models, since they were made from the essential task flow diagrams. We discovered that our essential task flow diagrams might have been too detailed, since this got us into trouble in the development of the dialogue models. Despite the problems with the inconsistency of the examples of the models, we found that the dialogue and presentation models provided an improved overview of the functions and how and which screens were to be implemented.

When transferring the interaction spaces model to a concrete GUI, the Wisdom method refers to the Bridge method, but several things were not taken into account by the latter method. First of all, there was no indication of how the grouping of objects should be done or how general design issues should be implemented. Even more important in our case, it did not take the screen size into account. Again we realized the importance of the first prototype, since it provided the basic ideas for the design.

Despite any criticism of inconsistence in the models used in Wisdom, they have proven useful in several aspects. They gave a good understanding of the system, when made, and we did end up with graphical user interface design that proved useful in the usability evaluation, a system that managed to support the users' tasks in connection with problem solving. Having said this it should also be noted that the process of moving from the interaction space model to the concrete GUI design seems a bit like pulling a rabbit out of the magician's hat. What Wisdom really helps to do is that through its many models it manages to facilitate the designers in determining what information should be available in the specific windows and what information should be available for user manipulation and a navigational overview. But Wisdom does not help to position, group, estimate whether the cognitive load is acceptable, or what modalities to use. These fundamental challenges in user interface design all need to be taken into account when designing the user interface and are also closely related to the information presented in the window

5.2 Prototyping

Wisdom proposes development and evaluation of prototypes between the different workflows in order to evaluate the result. We find this approach to be troublesome, since it contains some problems:

- Elaborating essential use cases and essential task flows, with the aim at avoiding premature design decisions, contradicts the development of prototypes at the end of the requirement workflow, as it forces the development team to make design decisions.
- The lack of descriptions on how to construct a prototype makes the scope and purpose of the prototype unclear.
- The purpose of developing a prototype seems irrelevant, when the method does not specify how the results from the evaluation of the prototype are utilized in the development process.

The transition from the first to the second workflow illustrates these problems. This transition is supposed to involve some kind of non-functional prototype. However, other than suggesting prototype development and test during this workflow, the Wisdom method was not that specific on these issues. First of all, the Wisdom method had focused on *not* making any GUI design decisions until now, and instead focusing on using essential use cases and task flow diagrams, which was a kind of a problem when designing a prototype. Furthermore it did not specify how the results of the prototype test should be utilized in the further development.

The usability of the implemented prototype was tested through field evaluation in cooperation with the employees, whom are the potential users of the mobile system, and through heuristic inspection performed by usability experts. The overall results of these evaluations showed us that the system was indeed usable, and that the employees all said that they would want to use the system, if it were fully implemented.

6 Conclusion

In this paper, we have presented experiences from an experimental evaluation of the Wisdom method. The focus has been on the extent to which this method supports design of the HCI aspects of a mobile device for communication.

On the overall level, the procedure worked well. The activities were relevant, and the results of different activities comprised a coherent design. The shortcomings of the method relate to specific activities, where we had to employ other techniques. In addition, the use of prototypes was recommended but the interplay between specifications and prototypes was not supported by the method.

At first Wisdom seemed attractive to us, because it gave the impression that the design workflow aimed at facilitating the transition from object models to a concrete user interface design. Having used the method, we can now say that we did not find this to be the case in our development process. The activity of making the concrete user interface design, based on object models, is rather inadequate, since the models created in the design workflow only facilitate the developers in understanding *what* information is required in a specific window. All other design decisions, which all have significant influence on the usability of the implemented system, are, in essence, still based on the experience of the developers who are involved in the specific design of the user interface.

Our results and conclusions are limited by the fact that we have employed a case-based research approach. We have only applied the method once, and we have only developed a system for a specific domain. Therefore our approach's applicability in relation to the development of mobile systems in other environments is unknown.

Acknowledgments

We are grateful to the employees and the head of the fuel department at Nordjyllandsværket, Bruno Andresen, for giving us the opportunity to use the fuel department as the case in our system development process.

References

1. Alrø, H., Kristiansen, M.: Do You See What I Said? Humans Face to Face. (in Danish: Kan du se hvad jeg sagde? Mennesker ansigt til ansigt). Holistic, Roskilde, 1996.
2. John M. Artim (2001). *Entity, Task, and Presenter Classification in User Interface Architecture: An Approach to Organizing HCI Practice.* In Object Modeling and User Interface Design, edited by Mark van Harmelen. Addison-Wesley, 2001.
3. Francesco Belotti, Riccardo Berta, Alessandro De Gloria and Massimiliano Margarone (2003). *Supporting Efficient Design of Mobile HCI.* Mobile HCI 2003, Udine, Italy.
4. Sara Berg, Alex S. Taylor, and Richard Harper (2003). *Mobile Phones for the Next Generation: Device Designs for Teenagers.* CHI2003, Fort Lauderdale, Florida, 2003.
5. Larry L. Constantine and Lucy A. D. Lockwood (2001). *Structure and Style in Use Cases for User Interface Design.* In Object Modeling and User Interface Design, edited by Mark van Harmelen. Addison-Wesley, 2001.
6. Bo Dahlbom and Lars Mathiassen (1993). *Computers in Context – The Philosophy and Practice of Systems Design.* NCC Blackwell, Cambridge, 1993.
7. Tom Dayton, Al McFarland, and Joseph Kramer (1998). *Bridging User Needs To Object Oriented GUI Prototype Via Task Object Design.* In User Interface Design – Bridging the Gap from User Requirements and Design, edited by Larry E. Wood. CRC Press LLC, 1998.
8. Anind K. Dey and Gregory D. Abowd (2001). *Towards a Better Understanding of Context and Context-Awareness.* CHI2000, The Hague, The Netherlands.
9. Mark D. Dunlop and Stephen A. Brewster (2002). *The challenges of mobile devices for human computer interaction.* Personal and Ubiquitous Computing, Volume 6, Issue 4, 2002.
10. Jan Gulliksen, Bengt Göransson, and Magnus Lif (2001). *A User-Centered Approach to Object-Oriented User Interface Design*. In Object Modeling and User Interface Design, edited by Mark van Harmelen. Addison-Wesley, 2001.
11. Mark van Harmelen (ed.) (2001). *Object Modeling and User Interface Design* Addison-Wesley, 2001.
12. Mark van Harmelen (2001). *Designing with Idiom.* In Object Modeling and User Interface Design, edited by Mark van Harmelen Addison-Wesley, 2001.

13. Harper, R. *People versus Information: The Evolution of Mobile Technolog.* Mobile HCI 2003, Udine, Italy.
14. William Hudson (2001). *Toward Unified Models in User-Centered and Object-Oriented Design.* In Object Modeling and User Interface Design, edited by Mark van Harmelen. Addison-Wesley, 2001.
15. Jepsen, L. O., Mathiassen, L. Nielsen, P. A.: Back to thinking mode: Diaries for the management of information systems development projects. Behaviour and Information Technology, Vol. 8, No. 3 (1989).
16. Masao Kakihara and Carsten Sørensen (2002). *Mobility: An Extended Perspective.* Thirty-fifth Hawaii International Conference on System Sciences (HICSS-35, Big Island Hawaii, ed. R. Sprague Jr. IEEE. 2002.
17. Jesper Kjeldskov, Connor Graham, (2003). *A Review of Mobile HCI Research Methods.* Mobile HCI 2003, Udine, Italy.
18. Steinar Kristoffersen and Fredrik Ljungberg (1999). *Mobile Use of IT.* Proceedings of the 19th Information Systems Research Seminar in Scandinavia, 1999.
19. Philippe Kruchten, Stefan Ahlqvist, and Stefan Bylund (2001). *User Interface Design in the Rational Unified Process.* In Object Modeling and User Interface Design, edited by Mark van Harmelen. Addison-Wesley, 2001.
20. Lars Mathiassen, Andreas Munk-Madsen, Peter Axel Nielsen, and Jan Stage (2000). *Object-Oriented Analysis & Design.* Marko Publishing, Aalborg, 2000.
21. Simon McGinnes and Johnny Amos (2001). *Accelerated Business Concept Modeling: Combining User Interface Design with Object Modeling.* In Object Modeling and User Interface Design, edited by Mark van Harmelen. Addison-Wesley, 2001.
22. Nuno Jardim Nunes (2001). *Object Modeling for User-Centered Development and User Interface Design: The Wisdom Approach.* Ph.D. Thesis, Universidade da Madeira, 2001.
23. Nuno Jardim Nunes and João Falcão e Cunha (2000). *Wisdom: A Software Engineering Method for Small Software Development Companies.* IEEE Software, 2001.
24. Nuno Jardim Nunes and João Falcão e Cunha (2001). *Wisdom - Whitewater Interactive System Development with Object Models.* In Object Modeling and User Interface Design, edited by Mark van Harmelen. Addison-Wesley, 2001.
25. Mark Perry, Kenton O'Hara, Abigail Sellen, Barry Brown, and Richard Harper (2001). *Dealing with Mobility: Understanding access anytime, anywhere.* ACM Transactions on Computer-Human Interaction (TOCHI), 2001.
26. Mary Beth Rosson and John M. Carroll (2001). *Scenarios, Objects, and Points of View in User Interface Design.* In Object Modeling and User Interface Design, edited by Mark van Harmelen. Addison-Wesley, 2001.
27. Alex S. Taylor and Richard Harper (2002). *Age-old practices in the 'New World': A study of gift-giving between teenage mobile phone users.* CHI 2002, Minneapolis, Minnesota.